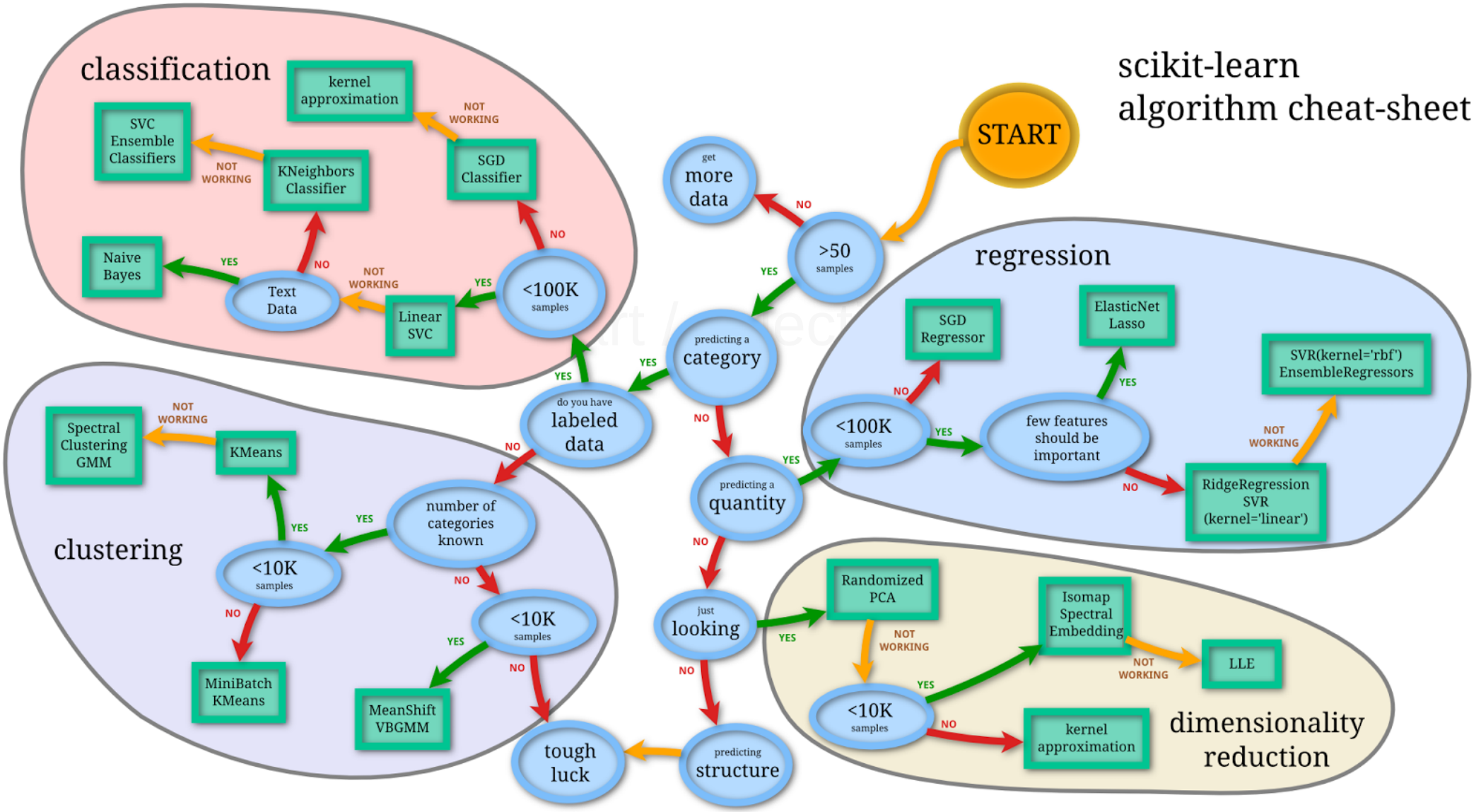
# Automatic Machine Learning?

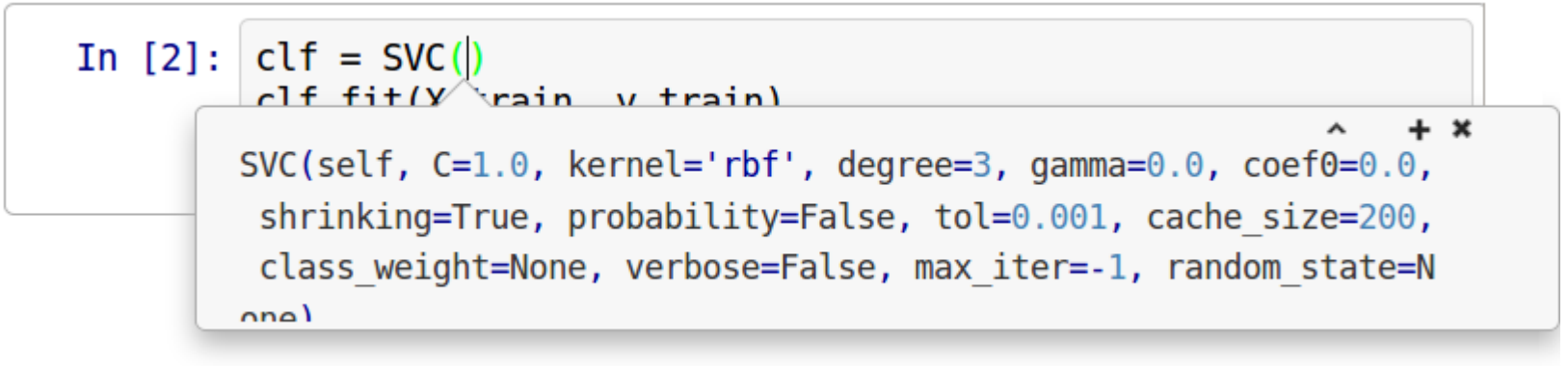

Andreas Mueller
(NYU Center for Data Science, scikit-learn)

Why?

# Issues with current tools (scikit-learn)


scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
predicting a category
YES
do you have labeled data
YES
<100K samples
NOT WORKING
Linear SVC
NOT WORKING
Text Data
YES
Naive Bayes
NO
KNeighbors Classifier
NOT WORKING
SVC Ensemble Classifiers
NO
SGD Classifier
NOT WORKING
kernel approximation
NO
classification
NO
predicting a quantity
YES
<100K samples
NO
SGD Regressor
YES
few features should be important
YES
ElasticNet Lasso
NO
RidgeRegression SVR (kernel='linear')
NOT WORKING
SVR(kernel='rbf') EnsembleRegressors
regression
NO
just looking
YES
Randomized PCA
NOT WORKING
<10K samples
YES
Isomap Spectral Embedding
NOT WORKING
LLE
NO
kernel approximation
dimensionality reduction
NO
predicting structure
tough luck
NO
do you have labeled data
NO
number of categories known
YES
<10K samples
NO
MiniBatch KMeans
YES
<10K samples
NO
MeanShift VBGMM
YES
KMeans
NOT WORKING
Spectral Clustering GMM
clustering

# Selecting Hyper-Parameters



# Scikit-learn: Explicit is better than implicit

```
make_pipeline(
    OneHotEncoder(),
    Imputer(),
    StandardScaler(),
    SVC())
```

# What?

```
from automl import AutoClassifier
clf = AutoClassifier().fit(X_train, y_train)
```

```
> Current Accuracy: 70% (AUC .65) LinearSVC(C=1), 10sec
> Current Accuracy: 76% (AUC .71) RandomForest(n_estimators=20) 30sec
> Current Accuracy: 80% (AUC .74) RandomForest(n_estimators=500) 30sec
```

# Step 1: Automate Parameter Selection

# Step 2: Automate Model Selection

# Step 3: Automate Pipeline Selection

How?

# Formalizing the Search Space

Discrete and Continuous Parameters
Conditional Parameters
Fixed pipeline vs flexible pipeline

# Formalizing the Search Space

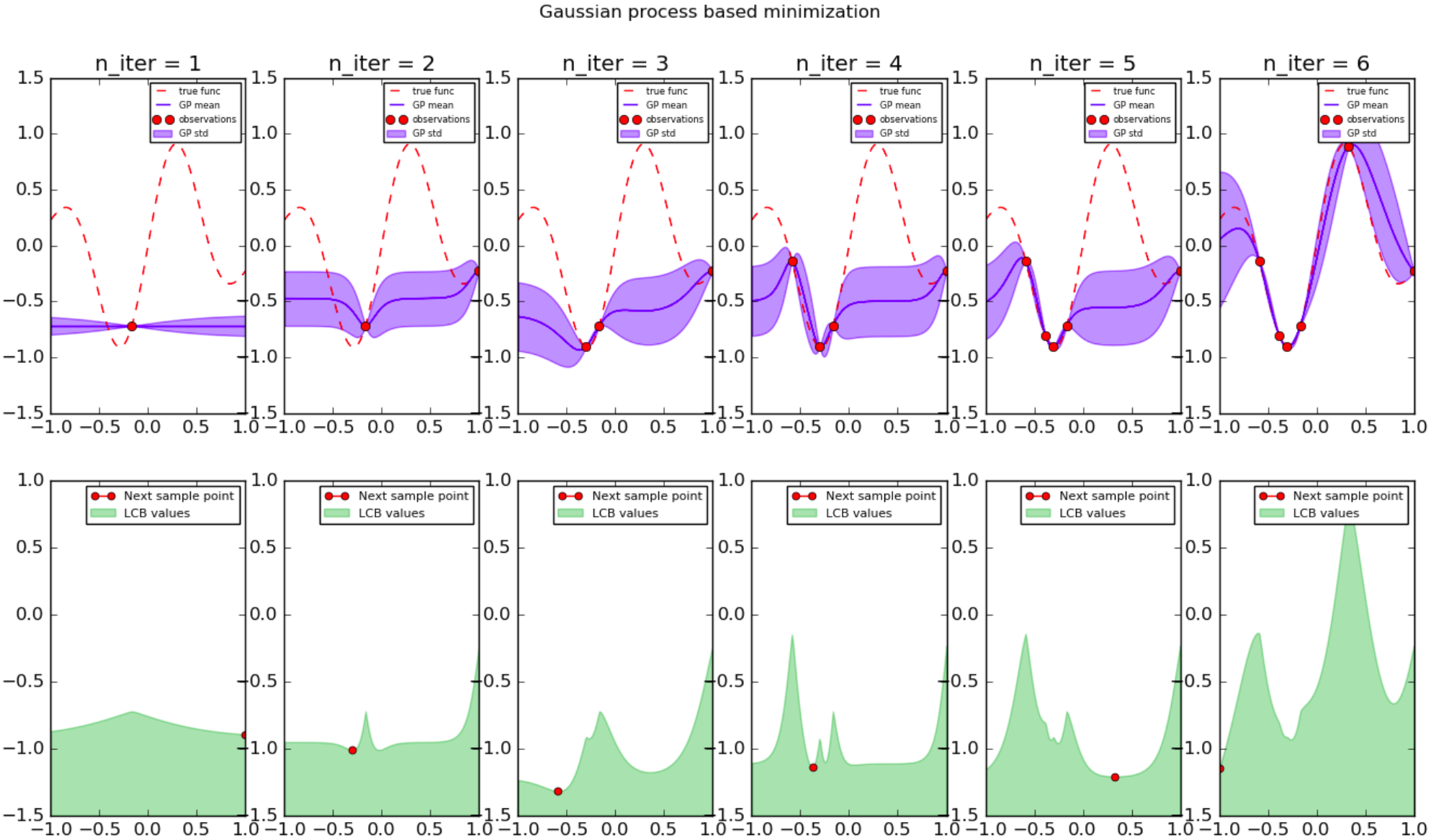
Discrete and Continuous Parameters
Conditional Parameters
Fixed pipeline vs flexible pipeline

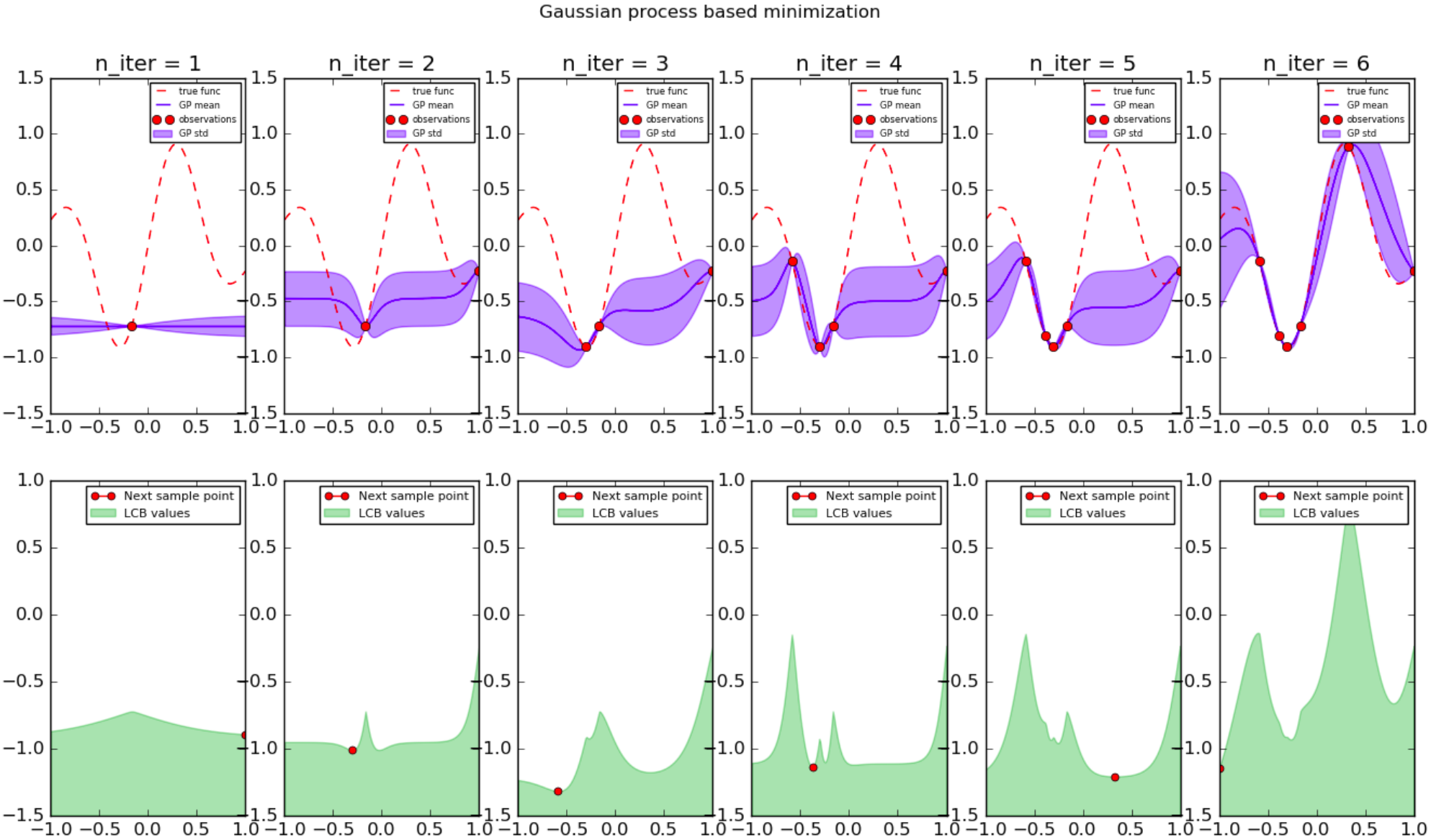
# Search Methods

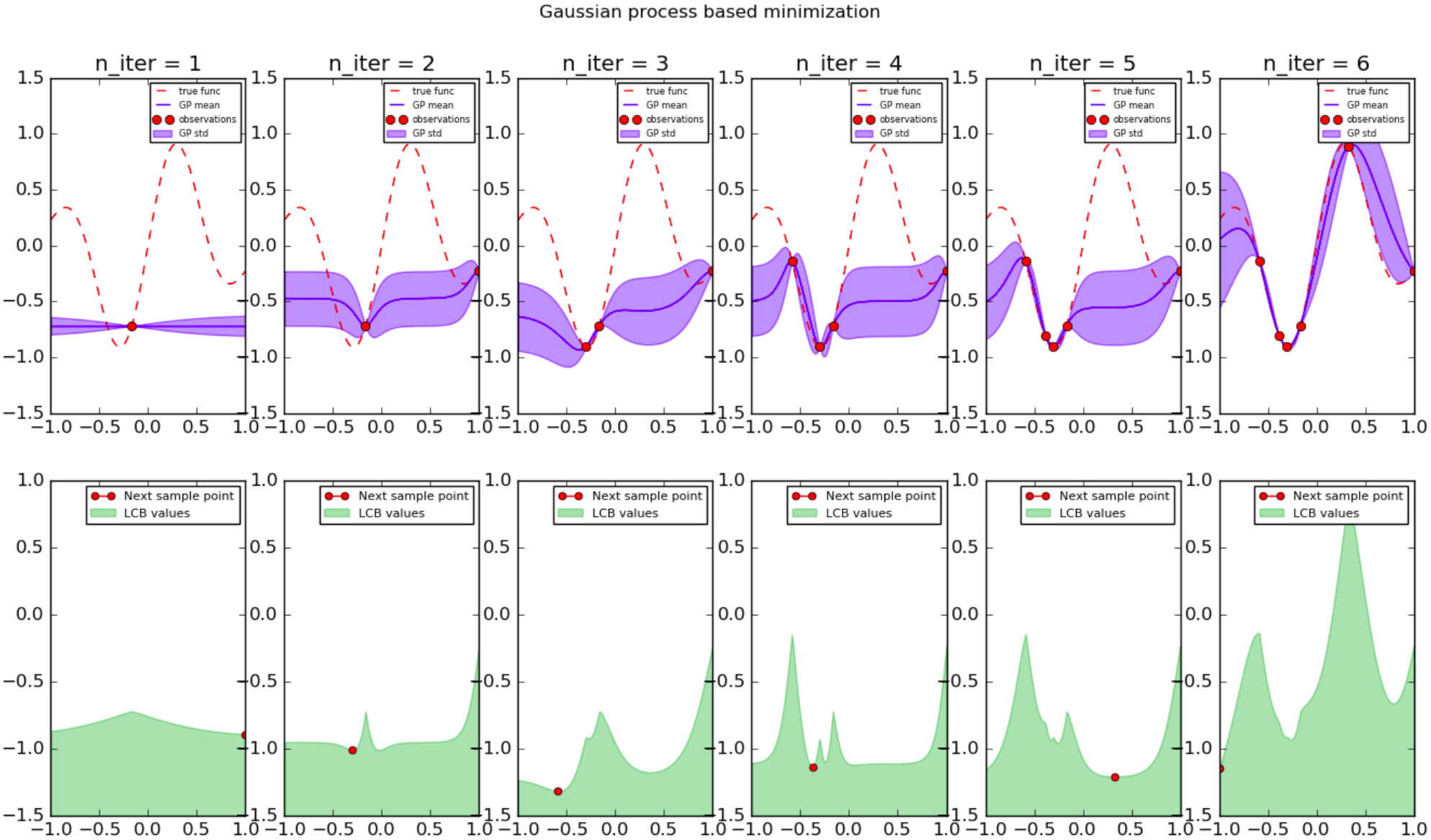
# Exhaustive Search (Grid Search)

# Randomized Search

# Bayesian Optimization (SMBO)


Gaussian process based minimization
n_iter = 1
n_iter = 2
n_iter = 3
n_iter = 4
n_iter = 5
n_iter = 6
true func
GP mean
observations
GP std
Next sample point
LCB values


Gaussian process based minimization
n_iter = 1
n_iter = 2
n_iter = 3
n_iter = 4
n_iter = 5
n_iter = 6
true func
GP mean
observations
GP std
Next sample point
LCB values

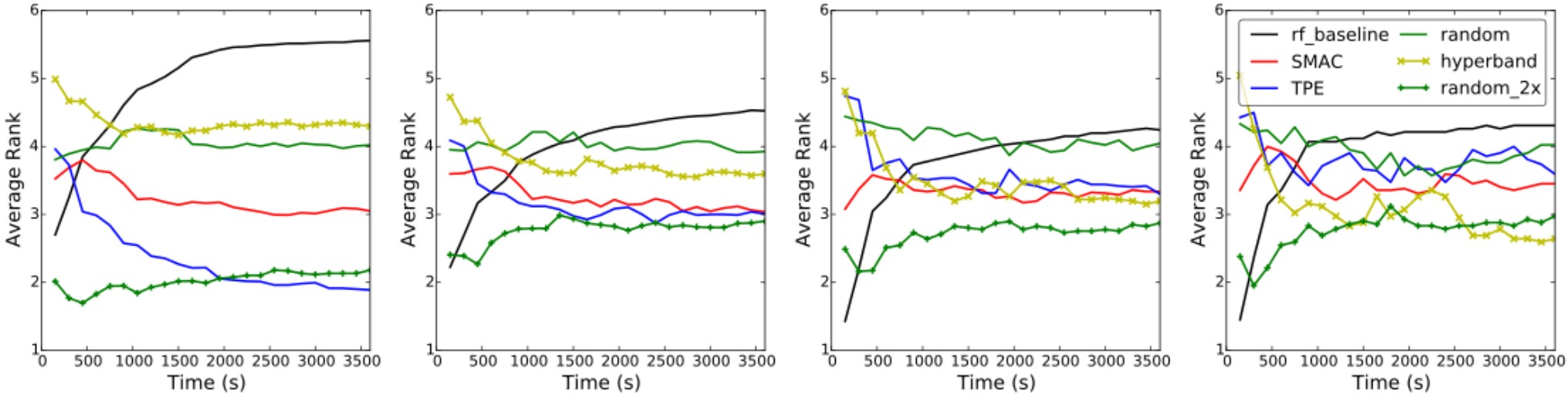

Gaussian process based minimization
n_iter = 1
n_iter = 2
n_iter = 3
n_iter = 4
n_iter = 5
n_iter = 6
true func
GP mean
observations
GP std
Next sample point
LCB values

# Gaussian Processes

# Random Forest Based (SMAC)

# Non-parametric (TPE)

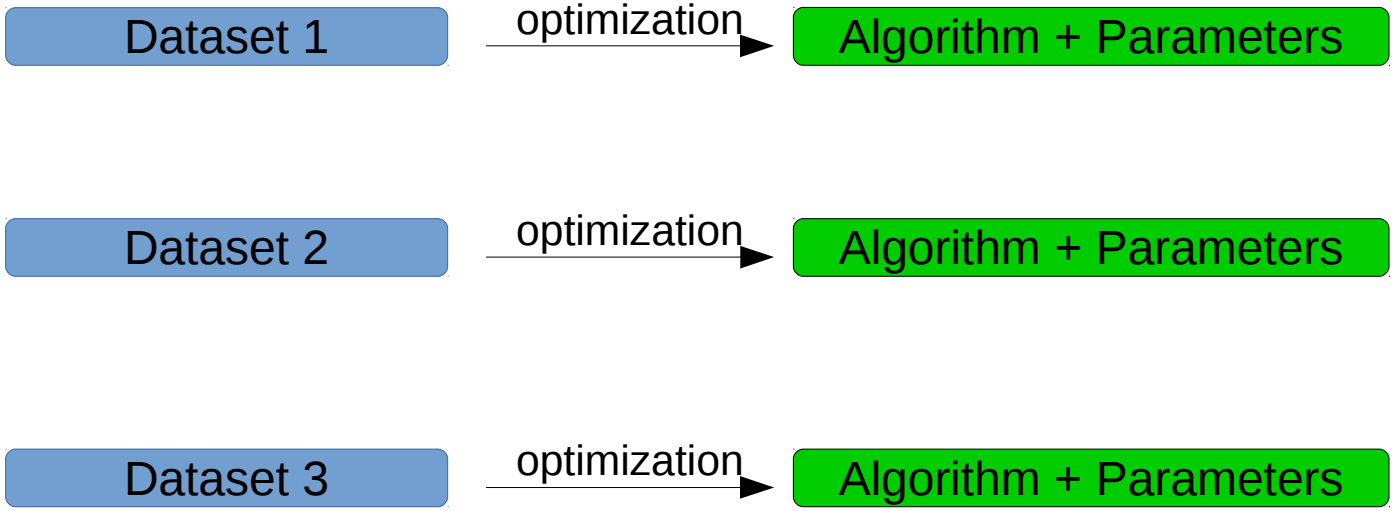
| Experiment | # Evals | SMAC | TPE | Spearmint | DNGO |
|---|---|---|---|---|---|
| Branin (0.398) | 200 | $0.655 \pm 0.27$ | $0.526 \pm 0.13$ | $\mathbf{0.398 \pm 0.00}$ | $\mathbf{0.398 \pm 0.00}$ |
| Hartmann6 (-3.322) | 200 | $-2.977 \pm 0.11$ | $-2.823 \pm 0.18$ | $\mathbf{-3.3166 \pm 0.02}$ | $-3.319 \pm 0.00$ |
| Logistic Regression | 100 | $8.6 \pm 0.9$ | $8.2 \pm 0.6$ | $\mathbf{6.88 \pm 0.0}$ | $\mathbf{6.89 \pm 0.04}$ |
| LDA (On grid) | 50 | $1269.6 \pm 2.9$ | $1271.5 \pm 3.5$ | $\mathbf{1266.2 \pm 0.1}$ | $\mathbf{1266.2 \pm 0.0}$ |
| SVM (On grid) | 100 | $\mathbf{24.1 \pm 0.1}$ | $24.2 \pm 0.0$ | $\mathbf{24.1 \pm 0.1}$ | $\mathbf{24.1 \pm 0.1}$ |


Average Rank
Time (s)
rf_baseline
SMAC
TPE
random
hyperband
random_2x
(a)
(b)
(c)
(d)

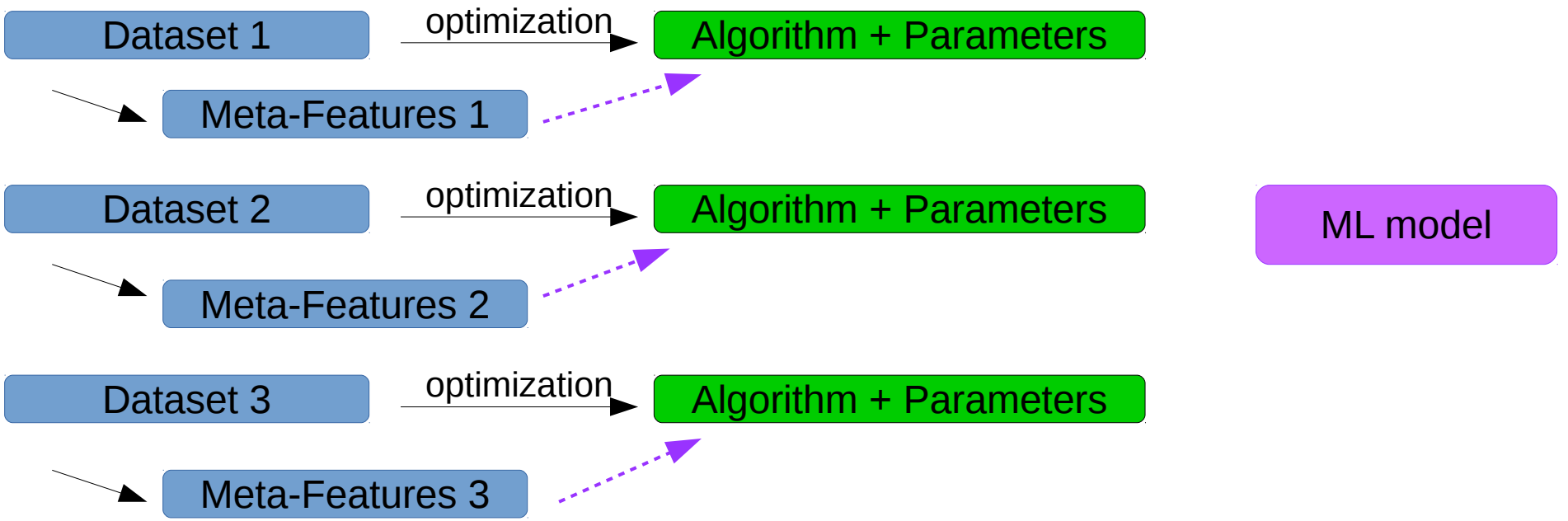
# Warm-starting and Meta-learning

# Meta-Learning

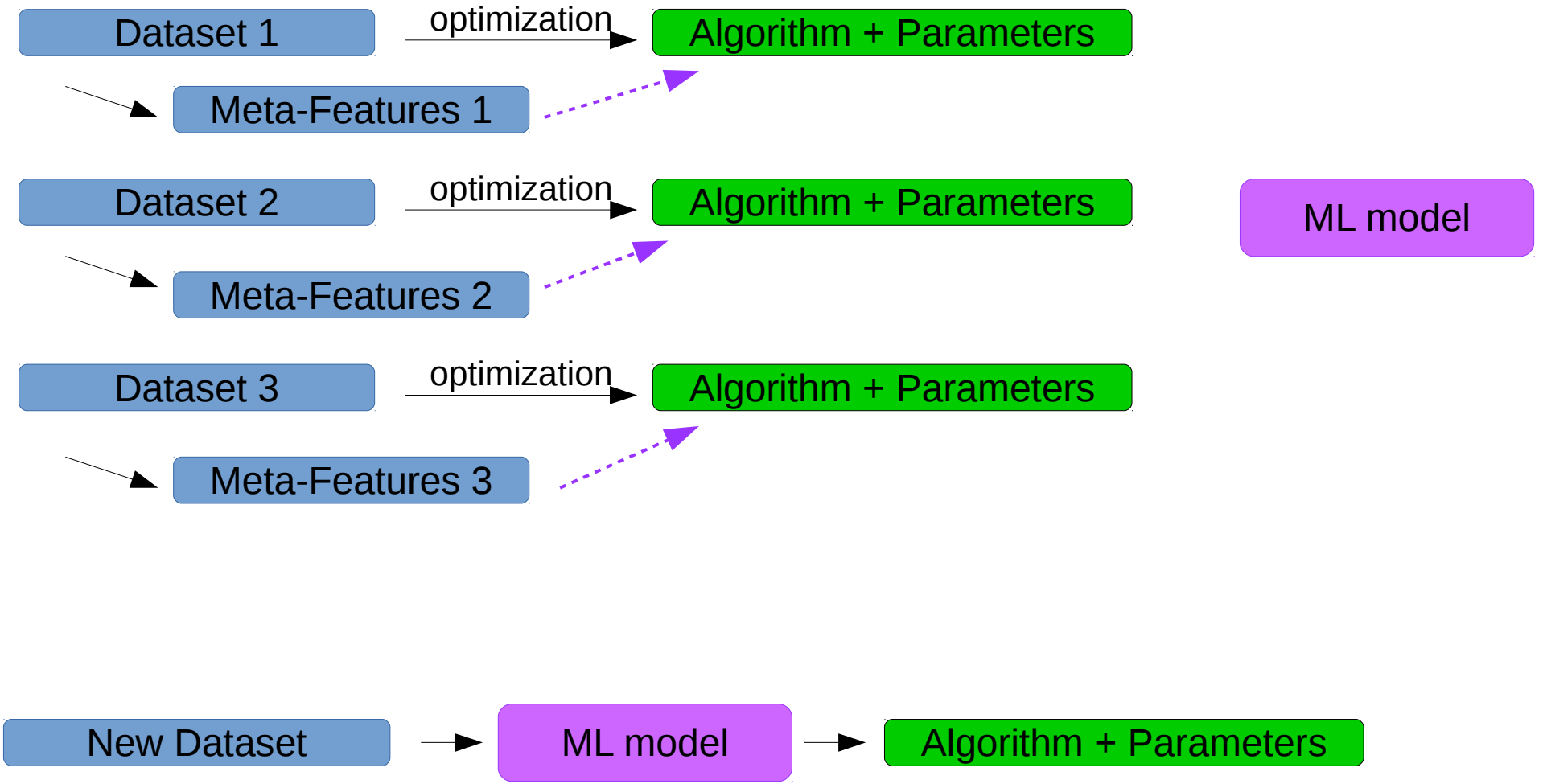
# Meta-Learning



# Meta-Learning



# Meta-Learning



# Meta-Features

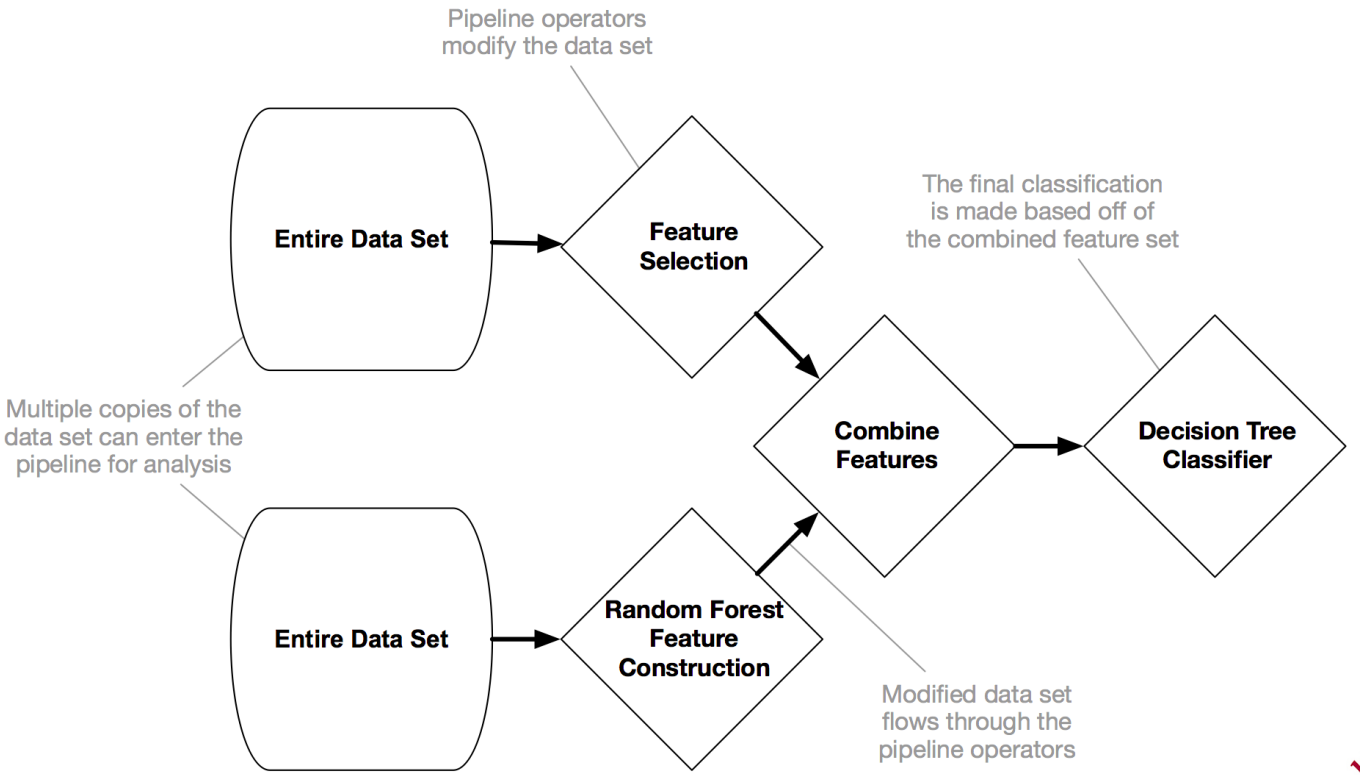
# Existing Approaches

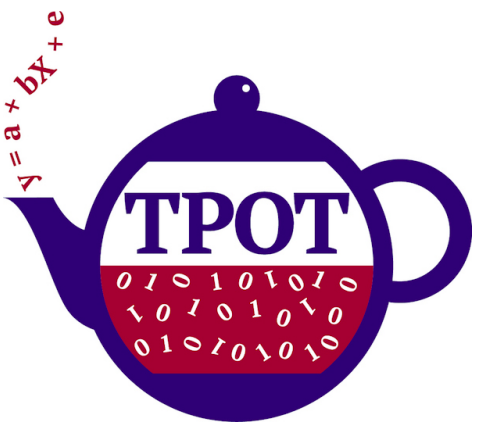
auto-sklearn
(Hutter, Feurer, Eggensperger)
http://automl.github.io/auto-sklearn/stable/

# Autoweka

# Hyperopt-sklearn

# TPot

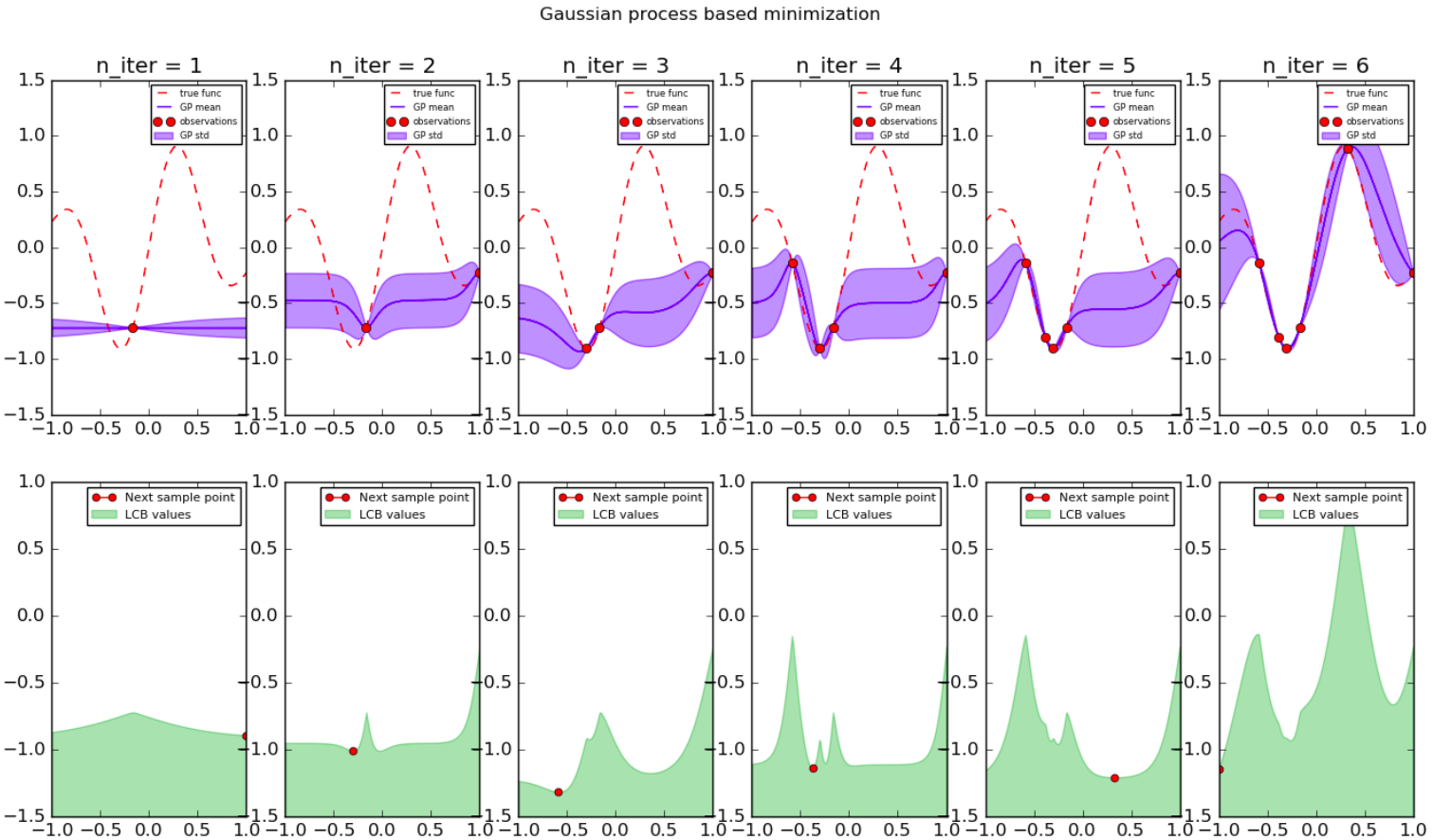

Pipeline operators
modify the data set
Entire Data Set
Feature
Selection
The final classification
is made based off of
the combined feature set
Multiple copies of the
data set can enter the
pipeline for analysis
Combine
Features
Decision Tree
Classifier
Entire Data Set
Random Forest
Feature
Construction
Modified data set
flows through the
pipeline operators


y = a + bX + e
TPOT

# Spearmint

https://github.com/HIPS/Spearmint

# Scikit-optimize



# Within Scikit-learn

- GridSearchCV
- RandomizedSearchCV
- BayesianSearchCV (coming)
- Searching over Pipelines (coming)
- Built-in parameter ranges (coming)

# TODO

Clean separation of:

- Model Search Space
- Pipeline Search Space
- Optimization Method
- Meta-Learning

- Exploit prior knowledge better!
- Usability
- Runtime consideration

# TODO

Clean separation of:

- Model Search Space
- Pipeline Search Space
- Optimization Method
- Meta-Learning

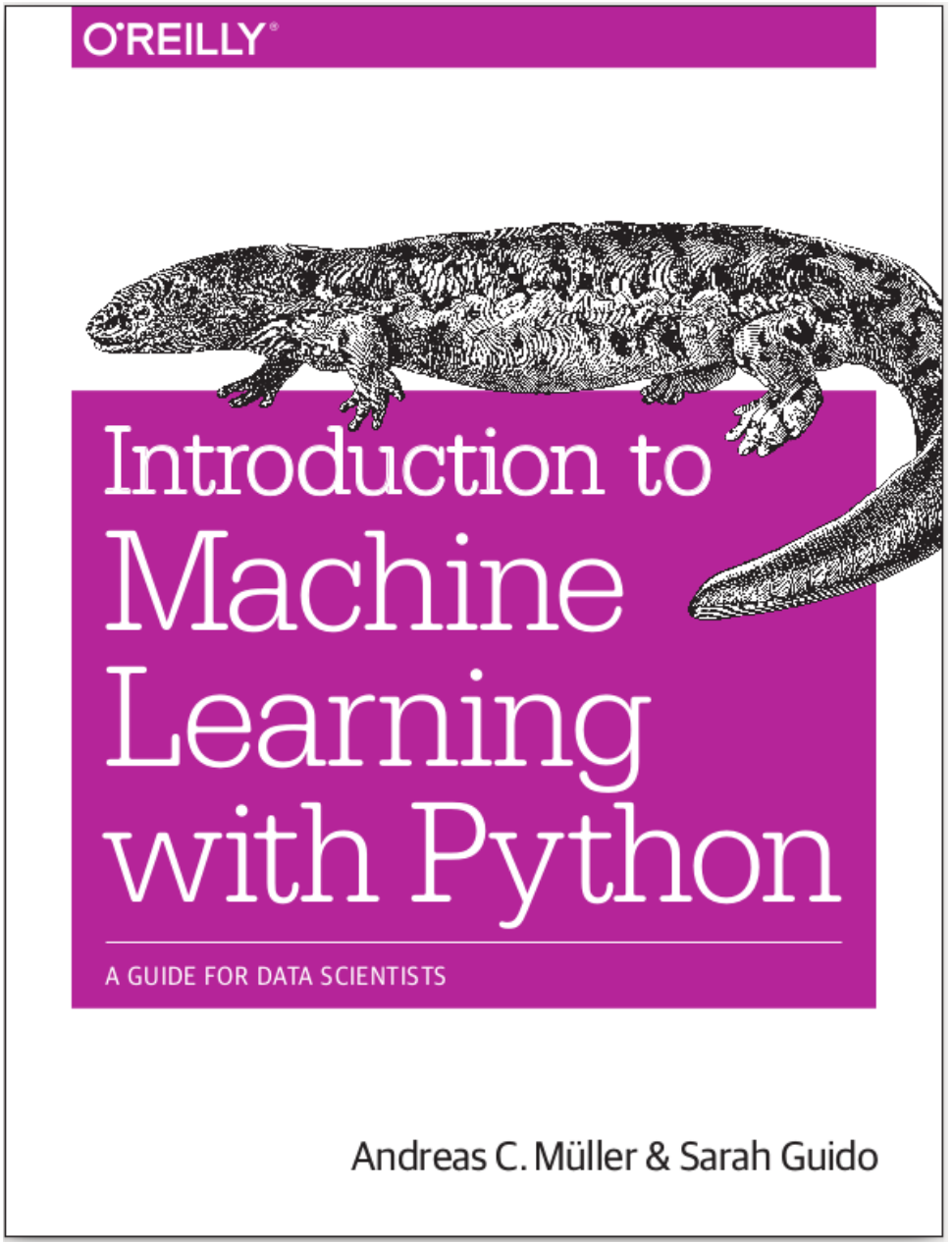
- Exploit prior knowledge better!
- Usability
- Runtime consideration
- Data subsampling

# Criticism

# Randomized Search works well

Do we need 100 Classifiers?
Do we need Complex pipelines?

I don’t want a black-box!




http://oreilly.com/pub/get/scipy

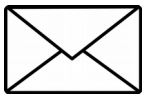
# Material

- Random Search for Hyper-Parameter Optimization (Bergstra, Bengio)
- Efficient and Robust Automated Machine Learning (Feurer et al) [autosklearn]
- http://automl.github.io/auto-sklearn/stable/
- Efficient Hyperparameter Optimization and Infinitely Many Armed Bandits (Lie et. al) [hyperband] https://arxiv.org/abs/1603.06560
- Scalable Bayesian Optimization Using Deep Neural Networks [Snoek et al]

# Thank you.


@amuellerml


@amueller


amueller@nyu.edu


http://amueller.io